Deterministic Optimal Control Analysis

Yuanxin Zhang (yz6201)

December 2022

1 Introduction

The creation of observers for control systems of the form is a significant issue in the study of nonlinear systems. The general form is

$$\dot{x} = f(x, u),$$

$$\dot{y} = h(x)$$

This constructed a filtering problem applied in various areas. Filtering refers to finding the state of a system from an incomplete and possibly noisy set of observations and a model which is also imperfect. Here we begin to review Mortensen's method of deterministic minimum energy estimation.[1]

$$\dot{x} = f(x(t)) + u(t), x(0) = x_0$$

$$\dot{y} = h(x(t)) + v(t), y(0) = 0$$

This is a system in which the initial condition x_0 is unknown, given an observation record, $Y_t = \{y(s), 0 \leq s \leq t, 0 \leq t \leq T\}$. This system creates the **synthetic data**.

And the cost function is

$$J_t(x_0, u) = S_0(x_0) + \frac{1}{2}\int_0^t (|u(s)|^2 + v(s)^2)ds$$

The purpose to build up the cost function is to minimize $J_t(x_0, u)$ with the minimum input (x^*, u^*, v^*) so that we can find the model that best fits the given data.

First, we replace $v(s)$ by using $v(t) = \dot{y}(s) - h(x(s))$ and omitting the $\dot{y}(s)^2$. The purpose to replace $v(s)$ is to formulate an unconstrained optimal control problem. And we omit $\dot{y}(s)^2$ here since this is an optimal question, but $\dot{y}(s)^2$ is a constant number, therefore, to simplify the calculation, we can directly omit this part.

Then we have

$$J_t(x_0, u, v) = S_0(x_0) + \frac{1}{2}\int_0^t (|u(s)|^2 + h(x(s))^2 - 2\dot{y}(s)h(x(s)))ds$$

So the deterministic energy estimate $\hat{x}(t)$ given Y_t is defined to be the endpoint of the optimal trajectory $s \to x^*(s)$ corresponding to a minimum energy pair $(x_0^*, u^*) := \hat{x}(t) = x^*(t)$.

The system above gives us an overview of finding deterministic minimum energy estimation. Now we turn to a specific question, with $\eta(s)$ as the given data

$$\dot{\eta}(t) = cos(\eta(t)) + e^{-t}$$

$$\eta(0) = 0$$

In the engineering application, η can be any synthetic data. For instance, it can be data assimilation for atmospheric predictability or prediction experiments.[2]

Meanwhile, u_1 and u_2 are two given control conditions.

$$u_1(t) = e^{-t}cos(t)$$

$$u_2(t) = \mathbb{1}_{[0,1)}(t) - 2\mathbb{1}_{[1,3)} + \frac{1}{2}\mathbb{1}_{[3,10)}$$

In this question, given an observation record $Y_t = \{y(s), 0 \leq s \leq t, 0 \leq t \leq T\}$ of the deterministic system with the initial condition of state x

$$\dot{x} = cos(x(t)) + u(t), x(0) = 0$$

$$\dot{y} = x(t) + v(t)$$

In the real application, the system is a simulation that needs to be verified by the given data η

compute the following cost functions. In this question, since we have given control $u(t)$ and given time periods, we only need to calculate the specific value of n_1 and n_2

$$n_1 = \frac{1}{2}\int_0^{10}(u_1(s))^2 ds + \frac{1}{2}\int_0^{10}|x_1(s) - \dot{\eta}(s)|^2 ds$$

$$n_2 = \frac{1}{2}\int_0^{10}(u_2(s))^2 ds + \frac{1}{2}\int_0^{10}|x_2(s) - \dot{\eta}(s)|^2 ds$$

(notation: we know that $|x(s) - \dot{\eta}(s)|^2$ is the norm of v)

2 Methods

For n_1 and n_2, we open the latter norm and get

$$n = \frac{1}{2}\int_0^{10}(u(s))^2 ds + \frac{1}{2}\int_0^{10}(x(s)^2 - 2x(s)\dot{\eta}(s))ds$$

In Mortensen's method of deterministic minimum energy estimation, η represents the given data, and our purpose to construct n is to minimize the cost so that it's more possible to reproduce the given data η, so we omit $\eta(s)^2$.; In this question, $\eta(s)^2$ are presented with specific values depending on the control function. For this purpose, the negligence of $\eta(s)^2$ almost doesn't influence the final result.

Then for the first part $\frac{1}{2}\int_0^{10}(u(s))^2 ds$, we can simply use staircase functions to simulate it. The codes can be seen in the Appendix with different $u(t)$. The calculation is based on

the principle of the Riemann Integral. The formula is

$$\int_a^b f(x)dx \approx \sum_{i=1}^{N} f(x_i')(x_i - x_{i-1}) = \frac{b-a}{N}\sum_{i=1}^{N} f(x_{i-1})$$

We divided the interval $[0, 10]$ into $N = 100000$ with $dt = 0.00001$. Then we regard the area of each square as parts of the integral and sum them up to get the Riemann Integral. The following graph shows the process.

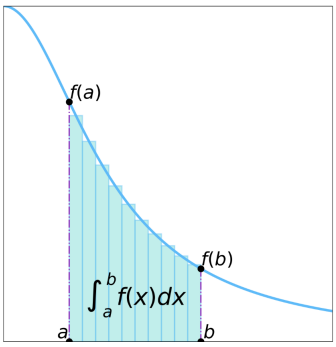

Figure 1: Riemann Integral

For the latter part $\frac{1}{2}\int_0^{10}(x(s)^2 - 2x(s)\dot{\eta}(s))ds$, here we use **Trapezoidal rule** to approximate the filtering problem for u_1 and u_2. The region under the graph of the function $f(x)$ is approximated to be a trapezoid in the trapezoidal rule, and its area is calculated. The formula is

$$\int_a^b f(x)dx \approx \sum_{k=1}^{N} \frac{f(x_{k-1}) + f(x_k)}{2}\Delta x_k$$

Here, $f(x) = x(t)$ and $f(x) = \eta(t)$. We still divide the interval $[0, 10]$ into $N = 100000$ with dt $= 0.00001$. The codes can be seen in the Appendix with different u. The following graph shows the Trapezoidal rule.

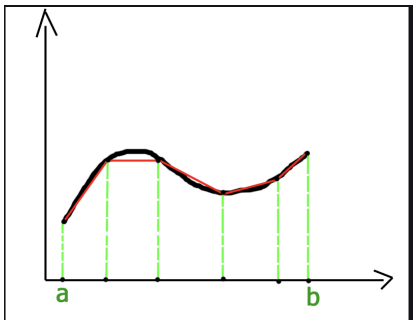

Figure 2: Trapezoidal rule

3 Results

After these calculations, we got the final result $n_1 = 9.8651$, and $n_2 = 15.5661$. The trajectories of $x(t), u(t), \eta(t), \dot{\eta}(t)$ with u_1 and u_2 are as follows.

- $x(t)$ represents the state of a system in terms of t
- $\eta(t)$ represents the synthetic data to simulate how the filtering works.
- $\eta der(t)$ represents $\dot{\eta}(t)$, which is the derivative of the synthetic data. The purpose to show this graph here is to calculate the latter part of the cost function, $\frac{1}{2}\int_0^{10}(x(s)^2 - 2x(s)\dot{\eta}(s))ds$
- $u(t)$ represents the trajectory of the control function.

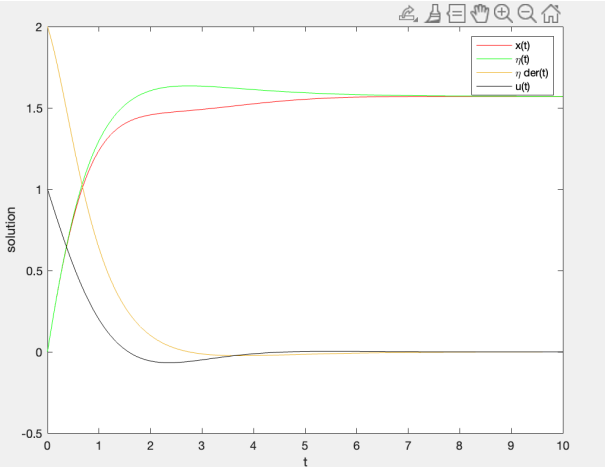

Figure 3: This is graph for $u_1(t) = e^{-t}cos(t)$

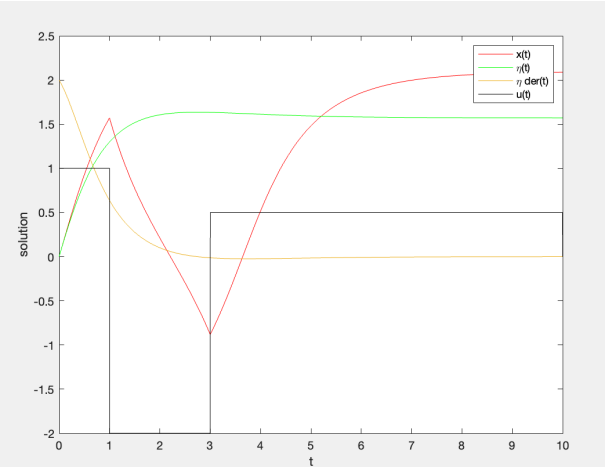

Figure 4: This is graph for $u_2(t) = \mathbb{1}_{[0,1}(t) - 2\mathbb{1}_{[1,3)} + \frac{1}{2}\mathbb{1}_{[3,10)}$

From these two graphs, we can compare the trajectory of $x(t)$ and $\eta(t)$. In figure 1, the trajectory of the two curves almost overlaps, while in the second graph, there is a huge difference between $x(t)$ and $\eta(t)$. Meanwhile, we compare the n value, and n_1 is much smaller than n_2. Therefore, we can conclude that the smaller n is (the smaller the cost function is), the more possible to reproduce the given data in real applications. The reasons to cause this result depends on the effect of control function $u(t)$ and $x(t)$. The entire process begins with the control function as given since these two control functions $u(t)$ are different (see

Figure 3 and Figure 4), two state functions $x(t)$ are different, therefore, the cost functions which are related to $u(t)$ and $x(t)$ will decide the cost from state $x_0 = 0$. Finally, we can use the cost function with HJB to find the minimum value depending on $x(t)$ and $u(t)$.

4 discussion

After the previous calculation, we use dynamic programming to study Mortensen's method of the deterministic minimum energy estimation problem (Bellman's equation). Our calculation doesn't show the dynamic programming part, but it's definitely our future work. Define a class of admission pairs (x_0, u)

$$U_{x,t} = \{(x_0, u) : x_u(0) = x_0, x_u(t) = x\}$$

which defines as pairs for which the corresponding trajectory passes through a specific point x at time t.

Then we define a value function

$$W(x,t) = \inf_{(x_0, u \in U_{x,t})} J_t(x_0, u)$$

This $W(x,t)$ is continuous and satisfies **the Bellman equation:**

$$\frac{\partial W(x,t)}{\partial t} + \hat{H}(x,t,DW(x,t)) = 0$$

$$W(x,0) = S_0(x)$$

where we have

$$\hat{H}(x,t,\lambda) = \max_{u \in U}\{\lambda(f(x)+u) - L(x,u,t)\}$$

Before we use the dynamic system to solve the problem, we know that $(x_0^*, u^*) := \hat{x}(t) = x^*(t)$ satisfies the minimum energy estimate. Therefore, we have the conclusion that

$$W(\hat{x}(t),t) \leq W(x,t) \text{ for all } x \in R^n$$

After using a series of theorem and lemmas (the basic idea is to prove W is both a viscosity subsolution and supersolution, this part will not be shown here). So $W(x,t)$ is the unique viscosity solution of the HJB Equation.

5 Appendix

This refers to the calculation of $\frac{1}{2}\int_0^{10}(u(s))^2 ds$ with u_1. Please see the codes here in detail.

```
% divide the interval as small as possible to improve
% the accuracy of the approximation.
dt = 0.0001;
N = 100000;
% integral for u_1(s)^2 between 1 and 10
u = zeros(N,1); t = u;
s = zeros(N,1);
for k = 1:N
    t(k) = k*dt;
    % we plug in the u_1 assumption
    u(k) = exp(-t(k))*cos(t(k));
    s(k) =u(k).^2*dt;
end
total1 = 1/2 * sum(s,"all");
```

This refers to the calculation of $\frac{1}{2}\int_0^{10}(u(s))^2 ds$ with u_2. Please see the codes here in detail.

```
% divide the interval as small as possible to improve
% the accuracy of approximation.
dt = 0.0001;
N = 100000;
% integral for u(s)^2 between 1 and 10
u = zeros(N,1); t = u;
```

```
s = zeros(N,1);
for k = 1:N
    t(k) = k*dt;
    % indicator functions expression
    if t(k)<1
        u(k) = 1;
    elseif t(k)<3
        u(k) = -2;
    elseif t(k)<10
        u(k) = 1/2;
    end
    s(k) =u(k).^2*dt;
end
total1 = 1/2 * sum(s,"all")
```

This refers to the calculation of $\frac{1}{2}\int_0^{10}(x(s)^2 - 2x(s)\dot{\eta}(s))ds$ with u_1. Please see the codes here in detail.

```
% TRAPEZOIDAL RULE for x(t) & \eta(t)
x = zeros(N,1); t = x;
x(1) = 0;
eta = zeros(N,1); t = eta;
eta_der = zeros(N,1);
eta(1) = 0;
eta_der(1) = 2;
for k = 2:N
    t(k) = (k-1)*dt;
    x_tmp = x(k-1);
    x_tmp_old = x_tmp;
    x_tmp = x(k-1)+dt/2*(cos(x(k-1))+cos(t(k-1))*exp(-t(k-1))+
            cos(x_tmp)+cos(t(k))*exp(-t(k)));
```

```
    x(k) = x_tmp;
    ss(k) = x(k).^2*dt;
    eta_tmp = eta(k-1);
    eta_tmp_old = eta_tmp;
    eta_tmp = eta(k-1)+dt/2*(cos(eta(k-1))+
            exp(-t(k-1))+cos(eta_tmp)+exp(-t(k)));
    eta(k) = eta_tmp;
    eta_der(k) = cos(eta(k))+exp(-t(k));
    result(k) = x(k).*eta_der(k)*dt;
end
total2 = 1/2 * sum(ss,"all");
total3 = sum(result,"all");
```

This refers to the calculation of $\frac{1}{2}\int_0^{10}(x(s)^2 - 2x(s)\dot{\eta}(s))ds$ with u_2. Please see the codes here in detail.

```
% TRAPEZOIDAL RULE for x(t) & \eta(t)
x = zeros(N,1); t = x;
x(1) = 0;
eta = zeros(N,1); t = eta;
eta_der = zeros(N,1);
eta(1) = 0;
eta_der(1) = 2;
for k = 2:N
    t(k) = (k-1)*dt;
    x_tmp = x(k-1);
    x_tmp_old = x_tmp;
    % indicator functions
    if t<1
        x_tmp = x(k-1)+dt/2*(cos(x(k-1))+1+cos(x_tmp)+1);
        x(k) = x_tmp;
```

```
    elseif t<3
        x_tmp = x(k-1)+dt/2*(cos(x(k-1))-2+cos(x_tmp)-2);
        x(k) = x_tmp;
    elseif t<10
        x_tmp = x(k-1)+dt/2*(cos(x(k-1))+1/2+cos(x_tmp)+1/2);
        x(k) = x_tmp;
    end
    ss(k) = x(k).^2*dt;
    eta_tmp = eta(k-1);
    eta_tmp_old = eta_tmp;
    eta_tmp = eta(k-1)+dt/2*(cos(eta(k-1))+exp(-t(k-1))+cos(eta_tmp)+exp(-t(k)));
    eta(k) = eta_tmp;
    eta_der(k) = cos(eta(k))+exp(-t(k));
    result(k) = x(k).*eta_der(k)*dt;
end
```

References

[1] M. James and John Baras. Nonlinear filtering and large deviations: A pde-control theoretic approach. *Stochastics An International Journal of Probability and Stochastic Processes*, 23:391–412, 03 1988.

[2] Jeffrey L. Anderson and Stephen L. Anderson. A monte carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741 – 2758, 1999.